Computing and Informatics, Vol. 26, 2007, 239–254

EXTENDING TEMPORAL ONTOLOGY WITH UNCERTAIN HISTORICAL TIME

Kamil Matoušek, Martin Falc, Zdeněk Kouba

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
166 27 Prague 6, Czech Republic
e-mail: {matousek, falc, kouba}@labe.felk.cvut.cz

Revised manuscript received 8 December 2006

Abstract. Temporal ontology for representing uncertainly specified time periods is presented. Mature approaches like Allen interval relations are combined with introduction of time granularity and time uncertainty concepts. The ontology is applicable both as a static data representation and for logical data inference. Logical conclusions can be derived using an automated inference system. Uncertainty parametrization was developed for handling the domain specific uncertainty characteristics. Temporal statements containing the most frequent expressions in the domain of cultural heritage preservation are identified and categorized with respect to their accuracy. A temporal inference system is implemented using OCML language. Consistency checks can find non-causal data clusters and lead to improving current event data. Finally, resource annotation with Dynamic Narrative Authoring Tool utilizing temporal inference is presented.

Keywords: Ontology, temporal reasoning, inference, historical periods

1 INTRODUCTION

A typical question in the historical domain asks, what artifacts are semantically related to the period of some event, like the time of reign of King Rudolph II. Typical cases, when such queries are of utility, are thematic exhibitions, where the objects that relate to some not exactly given period are to be selected.

Data from this field is often represented by records describing individual historical objects and stored in relational databases. The appearance of artifacts has to be recorded in detail for several reasons. Proper recognition and effective protection of artifacts provide just few of them. Being physical objects, they are located in space and time, while embedded in the social, history, and art context. Tracking changes of artifacts during time has always been a method that promised answering questions of art historians. In the domain of time, statements like "by the middle of the thirteenth century" or "during the reign of the King Charles the Fourth" are typical examples. Our contribution attempts to deal with uncertainty in temporal assertions. We aim at finding a suitable and effective inference mechanism, which would yield sufficiently accurate localization in time.

2 THEORETICAL FRAMEWORK

We start by building a theoretical framework for reasoning in the time domain. It combines the existing principles, like Allen relations, with time granularity and time uncertainty. Let us start with general specifications:

Temporal entity is the most general expression, which specifies a certain time. It is the root concept and common ancestor of other temporal elements.

(Finest) temporal scale defines all (finest) allowed temporal positions within a defined temporal domain. Temporal scale can be represented e.g. as an unambiguous mapping of temporal positions into real numbers. Originating position must be specified (e.g. zero on the scale equal to *January* 1st *1900 0:00:00 UTC* with the least interval one second). Temporal *position* is the numerical representation of temporal entities relative to a temporal scale.

(Simple) Time Point t is a named temporal entity located on the (finest) temporal scale. Time point is an instant. Time points have an attribute *location Loc(t)* of type *temporal position*. It can have maximum one value or it may be unspecified.

Temporal relation specifies relative positions of two temporal entities. For two time points, t_1 and t_2, the possible temporal relations are the following three:

$$t_1 \text{ before } t_2 \iff Loc(t_1) < Loc(t_2)$$

$$t_1 \text{ equals } t_2 \iff Loc(t_1) = Loc(t_2)$$

$$t_1 \text{ after } t_2 \iff Loc(t_1) > Loc(t_2).$$

They are mutually exclusive and one of them always holds for two simple time points.

Time quantity is a measurable amount of time, which can be expressed in terms of the temporal scale representing the distance of two time points. Similarly

as for distance, the definition allows only non-negative values of time quantities:

$$Q = |Loc(t_2) - Loc(t_1)|.$$

Time Interval $I(t_1, t_2)$ is a temporal entity which spans over multiple neighboring time points. Time interval can be represented as a pair of its endpoints, i.e. the starting and ending time points. The condition for all time intervals is that starting point is *not after* ending point, i.e.: $Loc(i_1) <= Loc(i_2)$. It is our choice to select the strict inequality in the condition. This way we will be consistent with the standards.

A property of a time interval is its *Duration* $Dur(I)$. It is the time quantity between its endpoints. If the locations of the time interval endpoints are specified, the time interval duration can be calculated as their difference:

$$Dur(I(i_1, i_2)) = Loc(i_2) - Loc(i_1).$$

Regardless specification of its endpoints, a time interval may also have its duration specified explicitly. Then consistency has to be checked according to the previous equation, which has to remain valid.

For a time point t and an interval $I(i_1, i_2)$, the possible temporal relations are:

$$t \text{ before } I,\ I \text{ after } t \iff Loc(t) < Loc(i_1)$$

$$t \text{ begins } I,\ I \text{ begun by } t \iff Loc(t) = Loc(i_1)$$

$$t \text{ during } I,\ I \text{ contains } t \iff Loc(t) > Loc(i_1)\ \&\ Loc(t) < Loc(i_2)$$

$$t \text{ ends } I,\ I \text{ ended by } t \iff Loc(t) = Loc(i_2)$$

$$t \text{ after } I,\ I \text{ before } t \iff Loc(t) > Loc(i_2).$$

2.1 Allen Relations

There are thirteen possible and mutually exclusive relations between two time intervals $I(i_1, i_2)$ and $J(j_1, j_2)$, called Allen's interval relations introduced in [1]:

$$I \text{ precedes } J \iff i_2 \text{ before } j_1$$

$$I \text{ meets } J \iff i_2 \text{ equals } j_1$$

$$I \text{ overlaps } J \iff i_1 \text{ before } j_1\ \&\ j_1 \text{ before } i_2\ \&\ i_2 \text{ before } j_2$$

$$I \text{ costarts } J \iff i_1 \text{ equals } j_1\ \&\ i_2 \text{ before } j_2$$

$$I \text{ during } J \iff j_1 \text{ before } i_1\ \&\ i_2 \text{ before } j_2$$

$$I \text{ cofinishes } J \iff j_1 \text{before} i_1\ \&\ i_2 \text{ equals } j_2$$

$$I \text{ equals } J \iff i_1 \text{ equals } j_1\ \&\ i_2 \text{ equals } j_2.$$

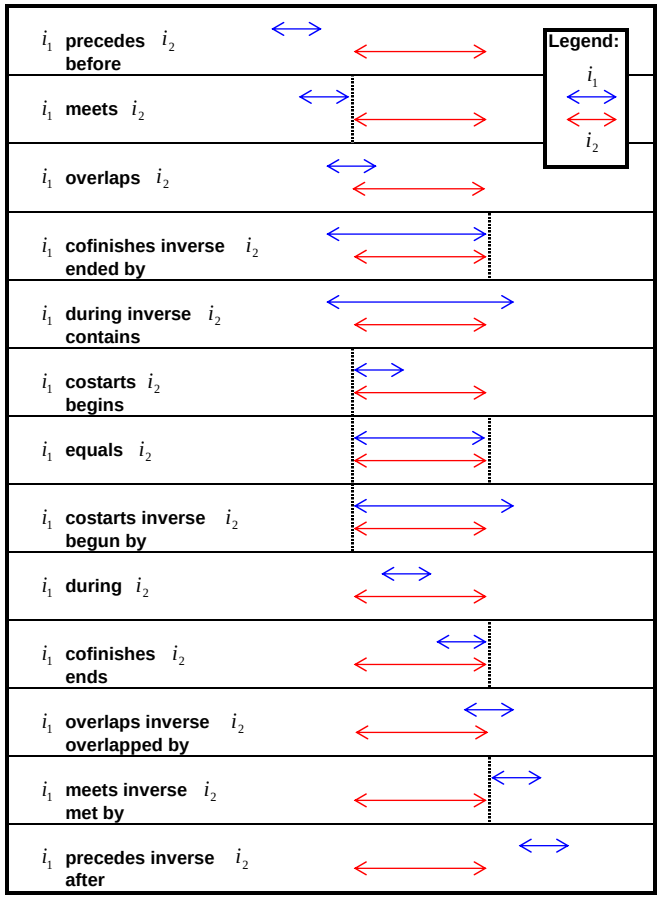

Fig. 1. Thirteen Allen's temporal relations of two time intervals

The remaining six inverse relations can be defined by swapping intervals I and J in the corresponding relations: I precedes inverse J, I meets inverse J, I overlaps inverse J, I costarts inverse J, I during inverse J, and I cofinishes inverse J. For graphical representation of the thirteen Allen relations see Figure 1 (moving from left to right corresponds to the passing of time).

2.2 Time Granularity

Time Granularity is the level of detail in which the time is considered (measured). The finest temporal scale defines *the finest granularity*. Different time statements can refer to different time granularity. E.g. "May, 12, 2003" is a time statement with *day* granularity, while "in 2002" has the *year* granularity. With coarser

time granularity their *granularity values* are defined, too. Time granularity defines its own unit scale for time positions.

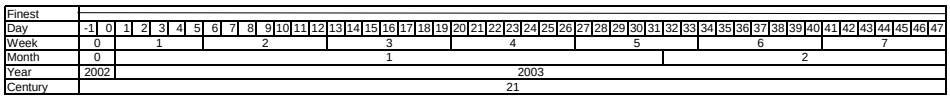

Fig. 2. Relationship of granularity temporal scales

Granularity temporal scale is a temporal scale, which orders all granularity values within a specified granularity.

Time Point t^g with Granularity g is a generalized time point. Its location can span over multiple (simple) time points in the finest temporal scale. So a time point with granularity is represented by the granularity value and its time granularity. Its *location* $Loc(t^g)$ is its representing time interval. On the other hand, function $Loc^g(t^g)$ assigns numerical *temporal position on granularity temporal scale* to a time point with specified granularity.

Time points with granularity, as well as simple time points, do not possess the property of duration. Their temporal property is to represent rather instants than intervals, but from a coarser perspective than simple time points. Granularity values are mapped into the finest temporal scale by transforming values of granularity into the *representing time interval* at the finest temporal scale.

$$I(i_1, i_2) = GR(t^g)$$

As a shortcut the following two functions directly return the locations of the endpoints of the representing time interval:

$$i_1 = Start(t^g), \qquad i_2 = End(t^g)$$

2.3 Uncertainty of Temporal Position

So far we have discussed the temporal representation of precise temporal statements either on the finest temporal scale or influenced by uncertainty using time granularity. Now we will go further into representation of other kinds of time uncertainty.

Time uncertainty represents an individual uncertainty type. Then *Uncertain Time Point* ${}^u t$ with *time uncertainty* u is a time point, which has its (uncertain) location and possibly spanning over multiple time points. Time uncertainty values appear explicitly as properties of uncertain time points.

The *location* $Loc({}^u t)$ of an uncertain time point is not given but *constrained*. However, in order to use the relative time uncertainty ranges, we need to have a reference position, which yields the *positioning context* of the time point. The amount of uncertainty is given by *Range of uncertainty of* ${}^u t$ $Range({}^u t)$, which is a temporal entity on the finest temporal scale, which includes all possible

temporal locations of a time point. Time uncertainty types, which we consider, fall into time intervals.

The suggested properties of time uncertainty are *FromTimePoint, ToTimePoint, BeforeRelTime* and *AfterRelTime*. Then representations of time uncertainty are:

1. For *absolute* specification of a *time range of uncertainty*, given by specified time points, we define uncertainty as a property of a pair of two time points: *FromTimePoint* and *ToTimePoint*.
2. For *relative time range of uncertainty*, there is another pair of relative temporal position properties and their respective granularity: *BeforeRelTime, AfterRelTime, BeforeGranularity* and *AfterGranularity*.

For such uncertainty values the representing time interval at the finest temporal scale is given by

$$I(i_1, i_2) = Range({}^{u}t).$$

Let us define two shortcut functions, which directly return the locations of the endpoints of the representing time interval:

$$i_1 = Start({}^{u}t) \qquad i_2 = End({}^{u}t).$$

Uncertain location $ULoc({}^{u}t)$ references the uncertain time point into a specific location on a temporal scale. To clarify the usage of this time uncertainty representation, let us give some examples: Temporal statements like "around the year 2002", "after 3^{rd} *of June 1990"*, "before the WWII", will use the *BeforeRelTime* and *AfterRelTime* together with *uncertain location* properties in its representation. On the other hand, if an event happened in some time "between the birth of the Emperor Charles IV and the death of the King Rudolph II", then *FromTimePoint* and *ToTimePoint* will be used.

For uncertain time points we cannot easily define common relations used to compare the finest time points like before, equals, after. Only sufficient conditions can be expressly corresponding to sufficient or necessary conditions:

$$\begin{aligned}
{}^{u_1}t_1 \text{ certainly before } {}^{u_2}t_2 &\iff End({}^{u_1}t_1) \text{ before } Start({}^{u_2}t_2)\\
{}^{u_1}t_1 \text{ certainly equals } {}^{u_2}t_2 &\iff Start({}^{u_1}t_1) \text{ equals } End({}^{u_1}t_1)\\
&\qquad \text{equals } Start({}^{u_2}t_2) mboxequals\ End({}^{u_2}t_2)\\
{}^{u_1}t_1 \text{ certainly after } {}^{u_2}t_2 &\iff Start({}^{u_1}t_1) \text{ after } End({}^{u_2}t_2)\\
{}^{u_1}t_1 \text{ possibly before } {}^{u_2}t_2 &\iff Start({}^{u_1}t_1) \text{ before } End({}^{u_2}t_2)\\
{}^{u_1}t_1 \text{ possibly equals } {}^{u_2}t_2 &\iff not(Start({}^{u_1}t_1) \text{ after } End({}^{u_2}t_2)\\
&\qquad or\ Start({}^{u_2}t_2) \text{ after } End({}^{u_1}t_1))\\
{}^{u_1}t_1 \text{ possibly after } {}^{u_2}t_2 &\iff End({}^{u_1}t_1) \text{ after } Start({}^{u_2}t_2).
\end{aligned}$$

These possible and necessary bounds of comparison relations fulfil the rule that if one time point is certainly before another, then also the former is possibly before the latter. It is easy to show that the condition for "possibly before" is weaker and subset of that of "certainly before" simply by applying the always valid requirement that:

$$not(End(^{u_1}t_1) \text{ before } Start(^{u_1}t_1)).$$

Of course, an analogous property holds for the remaining pairs: ⟨certainly equal, possibly equal⟩ and ⟨certainly after, possibly after⟩.

There is naturally a mapping of time points with granularity into their representation as uncertain time points. It is not possible to simply express by a general formula all time points with specified granularity as time points with a concrete uncertainty, because individual granularity values are mapped into the finest granularity via GR function, which need not be defined analytically by a formula.

2.4 Uncertainty Parametrization

Correct assessment of uncertainty types is often a problem. They are often domain dependent. Even the domain experts hardly define the imprecise terms of their daily use. That is why it is reasonable to suppose that individual instances of time uncertainty in a model of time can be parameterized. For example uncertainty *about a year* can be modeled as parameter, with e.g. initial value equal to twenty years in both directions before and after the referenced time. Later, e.g. after obtaining more relevant data, we may wish to modify the parameter value to correspond better to the real situation.

A challenging idea is to *"narrow"* the uncertainty by *"shrinking"* the existing model parameters in the way, which does not conflict with the facts that are known. Or, on the contrary, when facing inconsistency, uncertainty may be broadened to make the model consistent again. It corresponds to the notion of models in predicate logic, where we wish to have models, which are both consistent and complete. However, in general there is a whole space of possible models, which are consistent and complete. Through an iterative process we may wish to acquire the required consistency level (e.g. minimizing the uncertainty parameter space).

Let us have a *knowledge base KB* composed of domain instances. Let j be an instance from KB (e.g. an event) parameterized by an uncertainty parameter p_u:

$$j \in KB; j = j(p_u).$$

Let P denote the set of all uncertainty parameters in the knowledge base:

$$P = \{p_{ui} : (\exists j; j(p_{ui}))\}.$$

We wish to include all the KB instances in the current model composed of the ontology of temporal domain $ONT\text{-}T$ and a particular ontology of another domain

of interest *ONT-D*. Now we may wish to find optimal uncertainty parameters corresponding to the situation with the least uncertainty, which is required by the data:

$$\forall p_{ui} \in P : p_{ui}* = \min_c(p_{ui}),$$

where $\min_c$ denotes such minimal value of its argument, which still retains consistency, i.e. the following condition holds:

$$KB(P) \cup ONT\text{-}T \cup ONT\text{-}D \nvdash \bot,$$

where $\nvdash$ denotes *non-derivability*, i.e. impossibility to derive the consequence from given facts, and $\bot$ denotes contradiction. The question is whether a collection of restrictions in the knowledge base and ontology can be satisfied or not.

There are several possible realizations of an algorithm searching for the smallest parameter space. The algorithm can allow adjusting of one, several, or all parameters at the same time. Moreover, the parameters can be adjusted by a constant number or e.g. by a number proportional to the parameter value. Each of the uncertainty parameters has its finite initial value and its finite minimal allowable value (usually it can be zero). The algorithm starts with the set of uncertainty parameters P and an empty set Q in a discrete time t:

1. Check model consistency, if the model is inconsistent, return an error.
2. If P is empty, go to step 8.
3. Increase time $t := t + 1$.
4. Choose a parameter p from P and adjust its value of $p(t-1)$ to $p(t)$.
5. Check model consistency. If model is consistent and minimal allowable parameter value is not reached, go to step 2.
6. If the model is inconsistent, remove p from P and put the pair $\langle p, p(t-1)\rangle$ into Q, go to step 2.
7. If minimal allowable parameter value is reached, remove p from P and put the pair $\langle p, p(t)\rangle$ into Q, go to step 2.
8. Q contains resulting pairs of all parameters and their final values.

The proposed algorithm converges if the parameter adjustment in step 4 decreases the parameter value by a constant number, because then after a finite number of steps all minimum allowable parameter values would be reached. An optimal solution is found, if the parameters are mutually independent. A special case is represented by uncertainty using variable granularity. In this case the granularity parameter can represent directly an average length of the granularity unit, which can be adjusted.

An interesting property of some uncertainty specifications is their possible variation (growth) in the direction to the past as e.g. *"About 20 000 BC"* would certainly mean much bigger intended time than *"About the year 2001"*. In such cases

the parametrization should not depend on scalar parameter values, but on formulas. Even in this case it is possible to define parameters in these formulas and allow their modification as necessary. Of course, the selection of proper uncertainty parametrization is highly application and domain dependent.

3 CATEGORIES OF STATEMENTS

Regarding the individual expressions of time, there is a wide range of precise, imprecise, or uncertain artefact dating, which causes difficulties and further inaccuracy in any subsequent use of the data. So, assigning a value to a time property of an object does not mean that the object can simply be stuck to a defined position on a timescale. Uncertain data can have either inexact position on the timescale or inexact duration. The properties of time continuity and causality lead to the existence of implicit bindings of the time events and periods, which need to be respected while inferring some conclusions.

We have compiled a categorization of temporal statements containing the most frequent expressions in the domain of our interest with respect to their accuracy:

1. *Precise statements.* The whole data is available, maximum precision is reached (e.g. January 12, 2004, 12:30:00).
2. *Statements with higher granularity.* Data is available, but not so precise. It is necessary to distinguish instants and intervals (e.g. January 12, 2004 can be seen either as an instant of higher granularity or as a 24 hour time interval).
3. *Incomplete statements.* Some information is missing for precise time identification. One may intentionally use this kind of statement for recurring temporal positions – regularly repeated instants (e.g. January 12, 12:30:00).
4. *Uncertain statements with absolute specification of uncertainty* (e.g. between February 12 and February 13, 2004).
5. *Uncertain statements with relative specification of uncertainty* (e.g. around February 12, 2000, before 13^{th} century).
6. *Statements referencing other statements with temporal properties* (e.g. the period before the WWII, during the reign of the King Charles IV).
7. *Statements with unknown or missing information* (e.g. "that time I was ...").

Expressions related to the current time (e.g. yesterday, tomorrow) are supposed to implicitly belong to the category 6 as defined above.

4 TEMPORAL ONTOLOGY

Let us describe now a calendar and time system used in our temporal inference engine, which is able to model all the statement categories introduced in the previous section. We also provide a short description of a temporal reasoning system

implemented in OCML language [3]. It enables users to enter historical calendar data and to carry out temporal reasoning on a knowledge base containing temporal definitions.

4.1 Temporal Scale and Calendar Dates

Our temporal reasoning engine builds on the inference capabilities of OCML [3]. Its primary temporal coordinate system *(temporal scale)* is chosen to be that one internally built-in to the *Common LISP* language. It uses the temporal scale with zero point equal to *1. 1. 1900 0:00:00 UTC* and the shortest interval of one second. However, the extent of the LISP dates is limited. Thus, in order to offer calendar dates and times in an ancient history, it was necessary to introduce functions that decode and encode broader range of time positions.

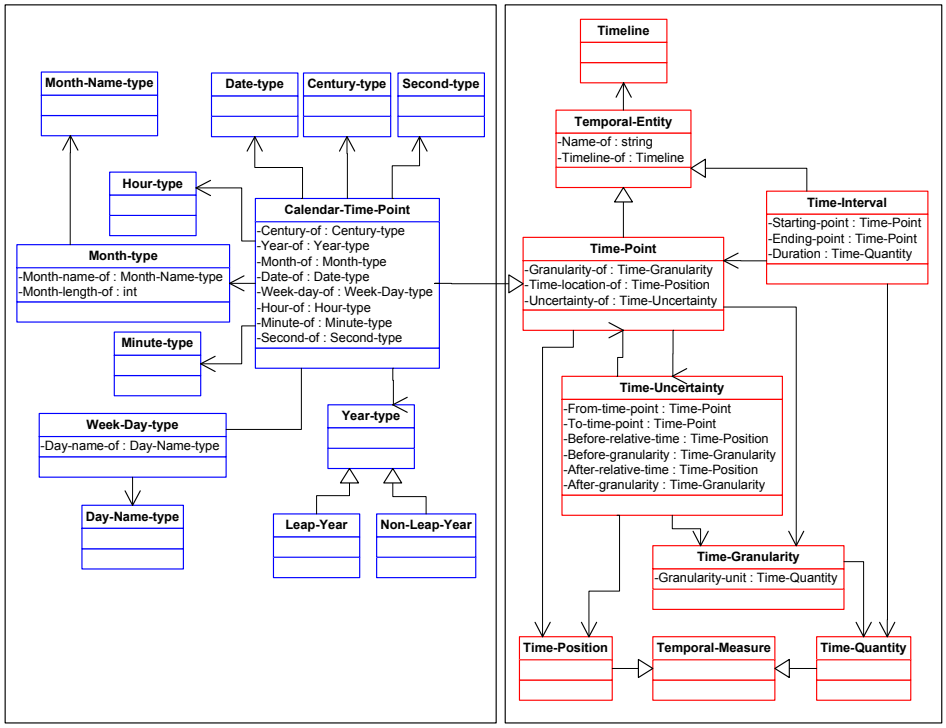

Fig. 3. Basic class structure

The basic class structure of the time ontology in our temporal inference engine using Unified modeling language is depicted in Figure 3. Classes in the right frame correspond to the definitions in our theoretical framework for reasoning in the time domain. The main components of our temporal model are subclasses of *temporal-entity* – *time-point* and *time-interval*. The property *timeline-of* of *temporal-entity*

enables distinguishing different kinds of temporal entities. Any instance of temporal entity can have any number of timelines assigned. Query results can be constrained by including the name of a timeline that is the subject of interest in the query submitted to the temporal inference engine. Thus, timelines define a kind of namespaces. Beside classes, the model contains basic time point and time interval relations, rules, and functions for time granularity and uncertainty manipulation.

Inspired by [4], in order to support calendar date and time specification, we introduced a subclass of *time-point*, denoted as *calendar-time-point* with the system of classes around it in the left frame of Figure 3. The slots *century-of, year-of, month-of, date-of, week-day-of, hour-of, minute-of,* and *second-of* enable representation of individual parts of calendar date and time. Currently, our model supports Gregorian dates with clock specifications.

4.2 Constraint Satisfaction

Some general constraints should always be satisfied, when working with temporal entities. One example is the property of transitivity of functions before and equals:

$$t_1 \text{ before } t_2 \ \& \ t_2 \text{ before } t_3 \Rightarrow t_1 \text{ before } t_3$$

$$t_1 \text{ equals } t_2 \ \& \ t_2 \text{ equals } t_3 \Rightarrow t_1 \text{ equals } t_3.$$

In order to prevent the model to become inconsistent, corresponding transitive closures have to be taken into account, e.g. via additional axioms. When adding new facts, corresponding constraints have to be checked.

4.3 Simple Practical Examples

Let us show how the temporal inference engine can be used. We model the reign of King Charles IV. At the beginning the important time points have to be defined.

```
(def-instance Charles-IV-start-reign Calendar-Time-point
    ( (date-of 26) (month-of 8) (year-of 1346)
    (granularity-of day-granularity)))
(def-instance Charles-IV-death Calendar-Time-point
    ( (date-of 29) (month-of 11) (year-of 1378)
    (granularity-of day-granularity)))
```

The following time interval is related to the specified time points:

```
(def-instance Reign-Charles-IV Time-interval
    (   (starting-point Charles-IV-start-reign)
        (ending-point Charles-IV-death)))
```

If a statement is inaccurate, we may need to create an instance of time uncertainty. Here, we model the birth of Socrates, who was born *around the year 470.*

It is possible to include uncertainty parametrization as in this case using *param-around-unc*:

```
(def-instance param-around-unc time-parameter((value-of 10)))

(def-instance Around-a-Year Time-Uncertainty

    (   (Before-relative-time param-around-unc)
        (Before-granularity year-granularity)
        (After-relative-time param-around-unc)
        (After-granularity year-granularity)))

(def-instance Sokrates-Birth Calendar-Time-point
    (   (year-of 470) (granularity-of year-granularity)
        (uncertainty-of around-a-year)))
```

Having defined the necessary facts, we may be interested in particular results using the inference engine. For example, having consulted the engine with data concerning all the periods of reign of Czech kings, the following query will retrieve the time interval corresponding to the Czech King ruling immediately after Ferdinand III:

```
(ocml-eval
   (findall ?a
       (and (timeline-of ?a Kings) (meets Ferdinand-III ?a))))
```

The returned result then finds King Leopold I: `(LEOPOLD-I)`

About thirty stories associated with South-Bohemian castles were annotated and evaluated. In two stories originated in two different castles, lord Oldřich of Rožmberk was mentioned. When presented by two different guides to a visitor of these two castles, the visitor might be confused assuming that both stories describe the same person. However, temporal inconsistency can be found in these two stories. The first story says that Oldřich of Rožmberk died in 1390. The second story describes Oldřich as a confirmed enemy of the Hussites. However, the Hussite movement rose as a consequence of burning Jan Hus at stake in 1415 after he had been accused of being a heretic. Clearly, this would be a contradiction in the visitor's mind as the Oldřich mentioned by the first story could not be the same person as the Oldřich who appeared in the second story. In this case, temporal reasoning performed on the set of semantic story annotations including representation of time will immediately discover the inconsistence.

5 DYNAMIC NARRATIVE AUTHORING TOOL

A tool called *Dynamic Narrative Authoring Tool* (DNAT) makes use of the temporal reasoning engine described previously. DNAT provides users with an integrated

narrative authoring environment designed to serve authors of knowledge intensive presentations, journal papers and educational materials. It also supports efficient organization of knowledge about both the domain of an emergent narrative and the narrative itself. Creation of semantic annotations of narratives for the use in *semantic web* is possible.

Several views of the same story can be created, which differ not just through writing or literary form but also through number of details incorporated in a story view – a *narrative*. A past event can be described including historical context within the borders of either world or regional history. DNAT supports different parallel series of historical events through its organization of events into timelines. DNAT is designed to communicate with the temporal inference engine.

Manipulation with events and their organization in timelines is possible by inserting and organization of knowledge in temporal events, story characters, etc. Multiple named timelines can be created or loaded. Data in the time domain is represented either as time points or time intervals, where both can be bound to an absolute temporal position or not. DNAT associates temporal data with events, because temporal information is often essential in stories. It helps to specify closely important moments like creation of objects, birth of characters, etc. Beside data purely in time domain, *temporal events* provide a general data structure intended for storing event type, event description, event location, and possibly an association with artifacts (i.e. images, video, etc.).

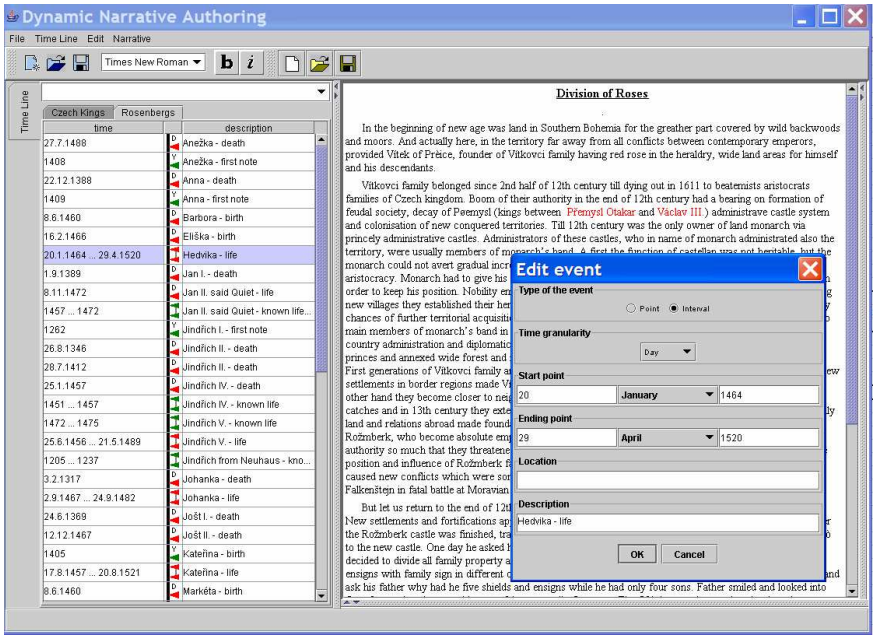

Fig. 4. Dynamic Narrative Authoring Tool

Temporal inference engine is used by a user to obtain the set of events corresponding to user's queries. During a session, user may ask questions, e.g.: *What happened in Bohemia during the governance of Charles IV?* Corresponding temporal query can be formulated using an interactive query builder and the inference module returns the resulting temporal events consistent with the temporal operator *during* and the defined temporal interval *governance of Charles IV*. Users can also pick all events that are important for a particular narrative of much wider story theme and compile them into a narrative timeline. Using simple drag & drop actions, event descriptions can be directly added to the emerging document.

6 CONCLUSIONS

We have presented our temporal framework and ontology for representing uncertainly specified time periods. Existing solutions included basic class structure with time-points and time-intervals so far, but they often neglected time uncertainty. We offer both absolutely and relatively specified uncertainty with respect to other temporal entities by combining Allen interval relations with time granularity and time uncertainty concepts. The preciseness of expressing temporal dates is captured and transformed in the form possibly processed by a computer system. Logical conclusions can be derived using an automated inference system possibly extended by uncertainty parametrization for handling the domain specific uncertainty characteristics and the algorithm for narrowing the parameter space.

In the domain of historical time, most frequent temporal expressions were categorized. Applications built on top of our temporal inference engine can provide ontology context-dependent information. A temporal inference system has been implemented using OCML language and modeling examples of the expression categories were showed. Modern collaborative technology using knowledge management approaches like in [2] can profit from the described historical temporal ontology and extend its own capabilities. An application example, Dynamic Narrative Authoring Tool utilizing the temporal inference engine, supports authoring of presentations, journal papers and educational materials by organizing and offering the needed resources.

Acknowledgement

The work has been supported by the research program No. MSM 6840770012 *Transdisciplinary research in the area of biomedical engineering II* of the Czech Ministry of Education, Youth and Sports.

REFERENCES

[1] Allen, J. F.: Maintaining Knowledge about Temporal Intervals. Communications of the ACM, Vol. 26, 1983, No. 11, pp. 832–843.

[2] Jung, J. J.: An Application of Collaborative Web Browsing Based on Ontology Learning from User Activities on the Web. Computing and Informatics, Vol. 23, 2004, No. 4, pp. 337–353.

[3] Motta, E.: Reusable Components for Knowledge Modeling. IOS Press, 1999.

[4] Zhou, Q.—Fikes, R.: A Reusable Time Ontology. Knowledge Systems Laboratory, Stanford University, 2000. Available on: `http://www.ksl.stanford.edu/KSL_Abstracts/KSL-00-01.html`.

Kamil Matoušek works as a scientist at the Department of Cybernetics, Czech Technical University in Prague, Faculty of Electrical Engineering. He is a member of the Gerstner laboratory at Czech Technical University and of its Knowledge-Based and Software Systems group. He reads lectures on Project Design and Management and Medical Informatics. He participated in several European as well as local projects. He has published papers in the field of data warehousing, data mining and knowledge engineering. His research interests include database systems, knowledge management, uncertainty modeling and processing, formal design of software systems as well as information systems in health service.

Martin Falc is currently finishing his Ph. D. studies at the Department of Cybernetics, Czech Technical University in Prague, Faculty of Electrical Engineering. His professional experience includes practical software design and development. He has published conference papers and research reports in the field of knowledge management. The topics of his research include ontology-supported knowledge management and formal design of software systems. He is a member of the Knowledge-Based and Software Systems group of the Gerstner laboratory at Czech Technical University.

Zdeněk Kouba is the Head of the Knowledge-Based and Software Systems group of the Gerstner laboratory at Czech Technical University in Prague. He reads lectures on Information Systems Design and Decision Support Systems. His research interests include database systems, uncertainty processing, object-oriented design and programming, and formal design of software systems. He is a (co)author of more than 50 publications in the above fields. He served as PC member and reviewer for several major journals and conferences. He has extensive experience in the EU research projects acting as the responsible Czech leader of many EU projects.